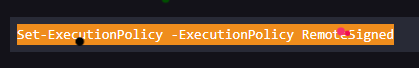
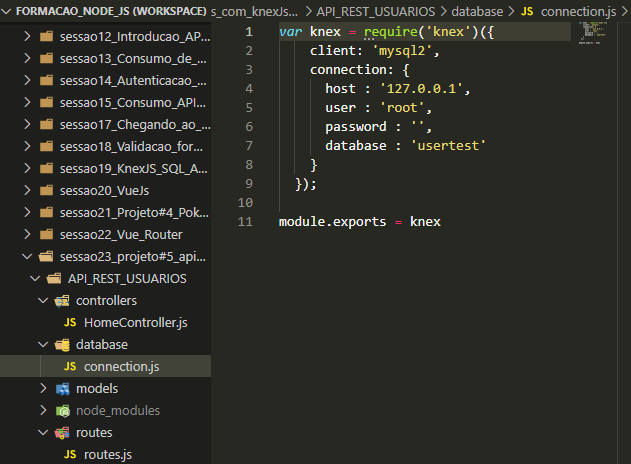
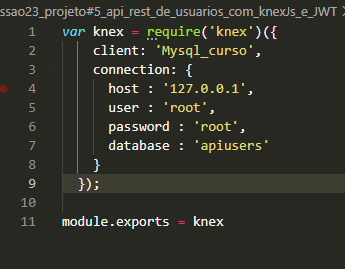
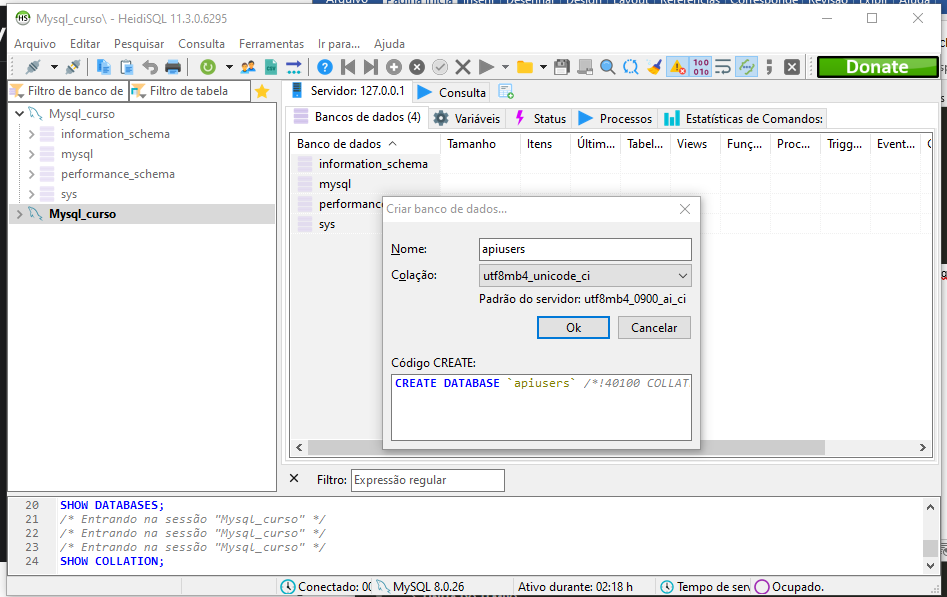
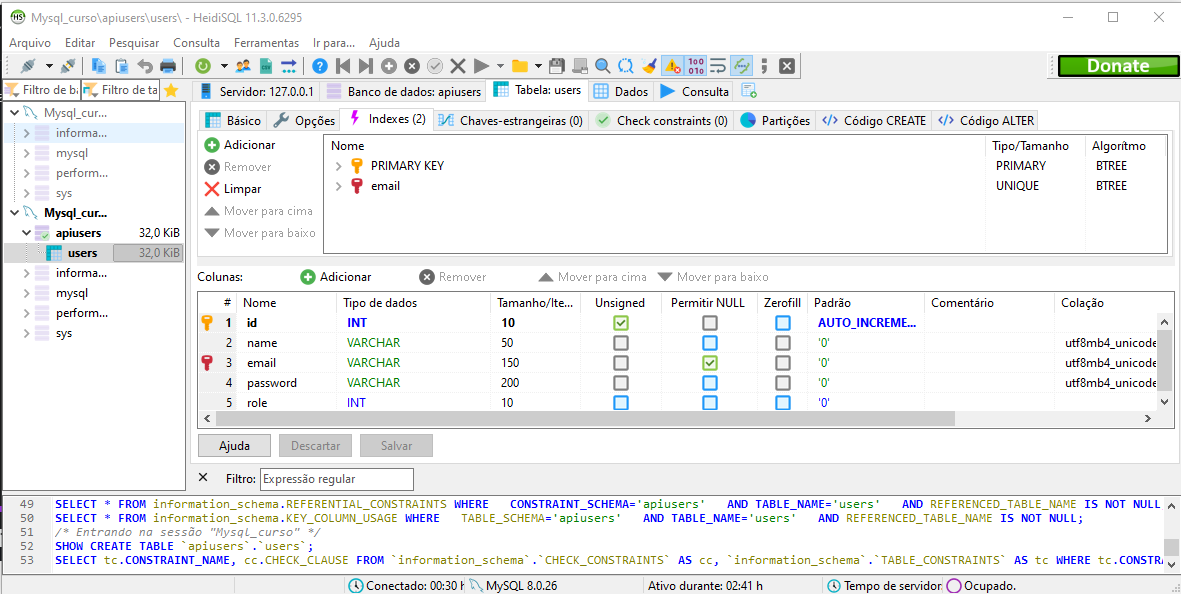
**SISTEMA DE GESTÃO DE USUÁRIOS**

1. Neste projeto iremos aprender a logica para implementação de um sistema de gestão de usuário completo, que pode ser utilizado em qualquer projeto futuro. Com cadastro, alterações, recuperação de senha etc...
2. Para agilizar a criação de um projeto foi criado a base de um projeto node no git hub para agilizar a criação
3. Link: github.com/techtuxbr/express-js-rest-api-base
4. Baixar projeto como ZIP, descompactar na pasta do projeto que será criado
5. Entrar na pasta do projeto e digitar : npm install
6. No arquivo index.js estamos importando quatro módulos
7. - boyParser que é utilizado para receber dados json via post.
8. – express
9. – router
10. Todas as rotas da aplicação serão definidas na pasta routes
11. Todos os códigos que se comunicam com o banco de dados será construído dentro da pasta models.
12. Para permitir código no powershell do Windows entre no powershell no modo administrador e altere a politica de segurança digitando:
13. 

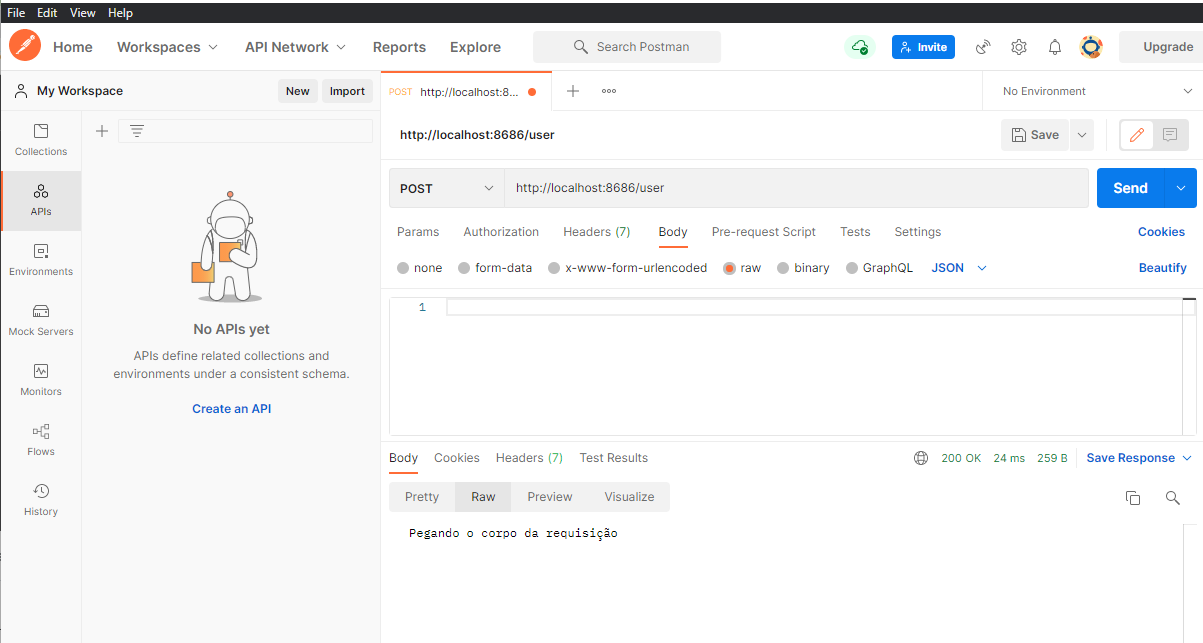
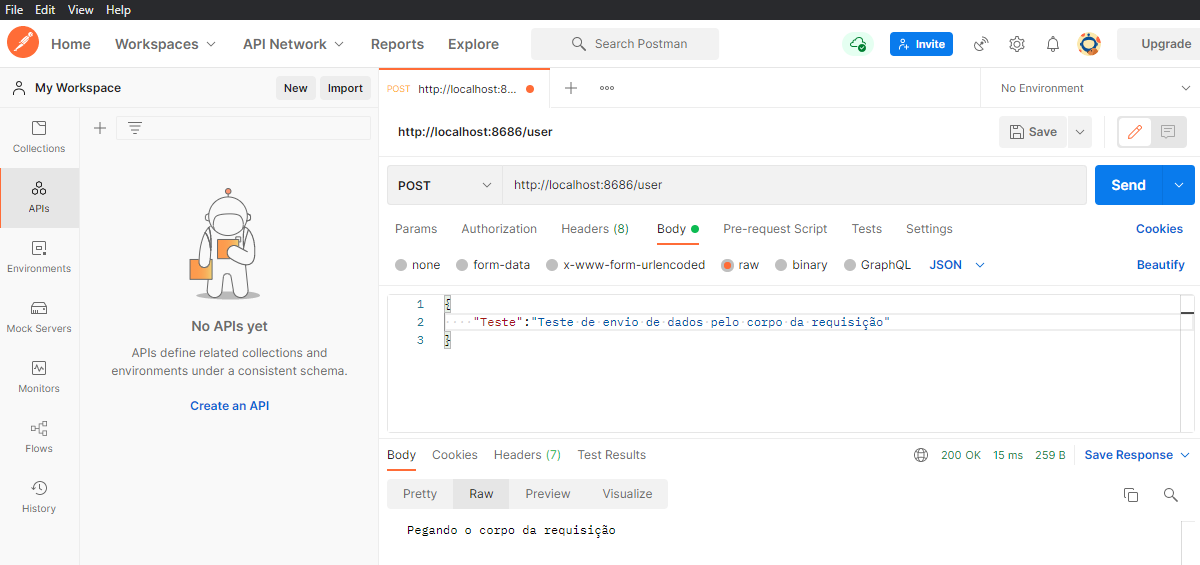
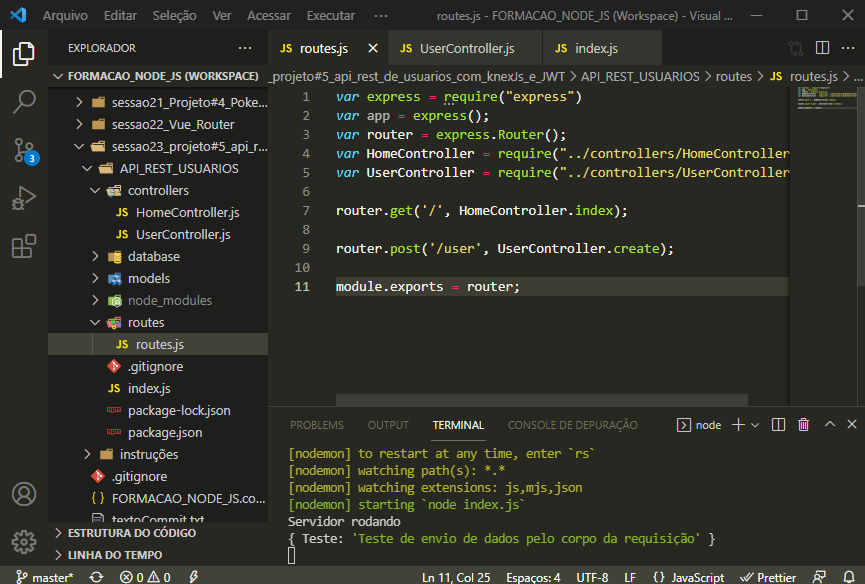
**Instalando o MySql**

1. Navegue até á a página mysql.com entre em Downloads, clique em “MySQL Community (GPL) Downloads”
2. Clique em “MySQL Community Server”
3. Selecione o sistema operacional e clique em “Go to Download Page”
4. Baixe a versão completa.
5. Após baixar clique no arquivo para instalar.
6. Quando abrir o programa, selecione a opção “Server Only” e next
7. Clique em “execute” para finalizar a instalação
8. Clique em next para iniciar a configuração.
9. Clique em next novamente
10. Primeiramente selecine “Develelopment Compute” em Config type.
11. Clieque em next
12. Cadastre um senha e next
13. Clique em next.
14. Clique em execute
15. Clique em finish
16. Clique em next
17. E Finish
18. Vamos linkar o executável do mysql no cmd
19. Vá até a pasta mysql em arquivos de programas
20. Entre na pasta bin
21. Copie o endereço desta pasta
22. Vá em painel de controle, sistema e segurança, sistema, propriedades do sistema, variáveis de ambiente,
23. Em variáveis de usuário para ...
24. Clicke em Path, editar, novo, colar o link que foi copiado, clique em ok , ok e ok.

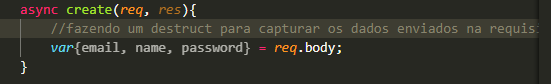
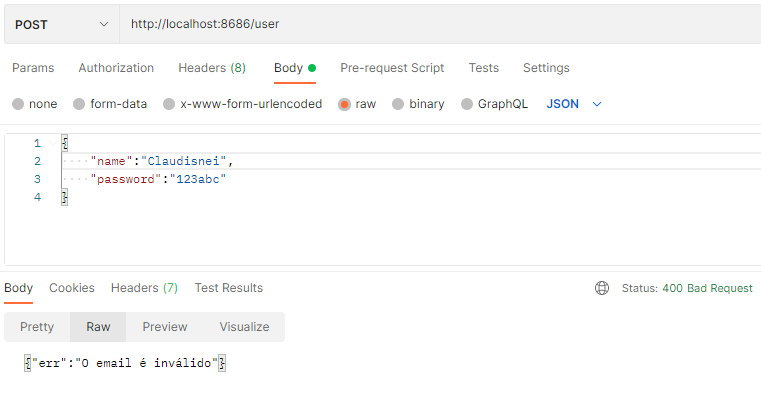
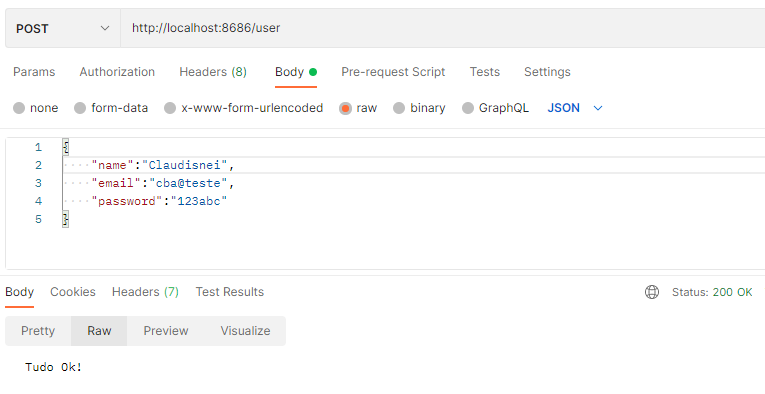
**Começando a crianção com o banco de dados**

1. Sempre que vc for criar uma aplicação com query builder como o Knex, temos que começar com a definição do banco de dados. Vamos utilizar o SGBD HeidiSql.
2. Antes vamos abrir o arquivo de configuração da conexão do banco de dados
3. 
4. Substituía os valores no arquivo connection, pelos dados de sua conexão com o mysql
5. 
6. Após isso vá no SGBD, neste caso no HeidiSQL e crie um banco de dados
7. 
8. Agora vamos definir a única tabela da api chamada users
9. Crie os campos :
10. 

**Se adaptando a estrutura dos Controllers**

1. na pasta controllers, crie o arquivo UserController.js para criar a lógica de acesso aos dados da aplicação
2. Lembrando que todo end point da aplicação deve ter uma resposta ‘res’
3. No fim do arquivo crie um module.exports para exportar o UserController().
4. 
5. Para usar o controller temos que importa-lo no arquivo de rotas e criar a rota em seguida. Neste caso vamos criar uma rota post já que vamos criar um novo usuário.
6. Router.post(‘/user’), o hast entende que como a rota é post e é passado somente o user, a rota será para salvar um novo usuário.
7. Para testar rotas post, temos que trabalhar om o Postman, por isso vamos instala-lo.
8. Após isso rode novamente o servidor
9. Vá no postman e digite a url <http://localhost:8686/user> na rota post.
10. Aparecerá a mensagem que configuramos no método create no UserController.
11. 
12. Para testar se os dados estão sendo enviados pelo corpo da páigna, no Postman, clique em Body, raw, selecione o formato Json. Escreva uma mensagem em Json. E clique em send.
13. 
14. No console do Vscode aparecerá a mensagem que foi enviada.
15. 

Validação de dados

1. Para evitar que o usuário envie dados desnecessários e incorretos para uma rota, os mesmos serão validados.
2. Já foi falado de validação de formulários no curso, por isso vamos fazer uma validação simples.
3. No método create do UserController vamos fazer um destruct, criando três variáveis que receberão os dados enviados pelo corpo da requisição.
4. 
5. Na sequência verificamos se o email é undefined, neste caso será enviado o status code 400 - bad request.
6. 
7. E caso contrário será enviado a resposta de status code 200 - ok
8. 
9. O código ficará assim:
10. 